Proactive business process mining for end-state prediction using trace features

Zeeshan Tariq
School of Computing
Ulster University
Jordanstown, UK
zeeshan@ulster.ac.uk

Darryl Charles
School of Computing
Ulster University
Jordanstown, UK
dk.charles@ulster.ac.uk

Sally McClean
School of Computing
Ulster University
Jordanstown, UK
si.mcclean@ulster.ac.uk

Ian McChesney
School of Computing
Ulster University
Jordanstown, UK
ir.mcchesney@ulster.ac.uk

Paul Taylor
Applied Reseach
BT
Ipswich,UK
paul.n.taylor@bt.com

***Abstract*—Business processes in the complex real-world environment are heterogeneous and challenging to monitor for any possible discrepancies. Businesses substantially rely on the efficiency of these processes to maintain the quality of services for their customers and wish to ensure that an executing business process is progressing in the desired manner. Although process mining techniques provide adequate information about the process execution, it is vital to maintain the quality of business processes through an automated process prediction system that analyses and provides constructive feedback for process improvement. Techniques in the literature can predict the future outcome of a business process, but they lack empirical information about the behaviour of an executing process instance as compared to the optimum process model. In this paper, we have proposed an online process prediction framework using features generated through process mining techniques. We used a heuristic miner algorithm to discover the process model and performed conformance analysis to generate features presenting the contextual behaviour of the process instance. We selected highly contributing features to predict the outcome of the real-world business process using several machine learning algorithms. Our experimental results showed high accuracy, recall, and F-measure. We compared our technique with a similar technique from literature and showed that our solution is more reliable in process outcome prediction.**

***Index Terms*—Business processes, Process Mining, Conformance Analysis, Feature Engineering, Process Prediction.**

I. Introduction

In recent times, organizations have increasingly competitive in achieving their business compliance. Also, there has been a continued interest in the research for improving business processes, especially for forecasting business process outcomes and steps to be taken to mitigate possible faults in process execution. For this, the business must estimate the likely outcomes of the business process in advance with high accuracy, as this could significantly impact the customer's satisfaction [1]. Early prediction of the outcome of business processes can effectively make business processes more cost-efficient by increasing customer satisfaction, decreasing churn out rate and improving the process execution time. The time required to complete a business process is a cumulative execution time of all the activities performed within a unique process instance [2]. Predicting the anticipated path of an executing business process helps businesses to engage with their customers at runtime, plan early to avoid possible delays and manage customers' expectations in advance [3]. Early detection of business process discrepancies may also assist in anticipatory steps to achieve the desired process outcome. A business process is a set of tasks/events carried out by an organization to achieve business goals, starting with a common entry point and generally following predefined events. The process can be formally defined as a sequence of discrete tasks, referred to as events. $P = \{s^{(i)}\} = \{s^{(0)}, s^{(1)}, .., s^{(n)}\}$. Generally each of such events $s^{(i)} \epsilon S = \{s_1, ..., s_N\}$ is a labelled and timed array of event parameters that characterize the event or task: $s^{(i)} = ti, [x_1^{(i)}, ..., x_m^{(i)}] = \{ti, x^{(i)}\}$. Each event starts at certain time referred as ti and finishes at $ti{+}Ti$, such that Ti is the total duration of the event. Each unique process instance is termed as a *trace*, so in each trace events occur at distinct times, i.e. $\forall_{i\epsilon 1,...n} ti + T_i \leq t_{i+1}$.

Process mining refers to a variety of techniques for analysing business processes through actual event logs generated as a result of the process execution. It is especially beneficial when there exists no standard definition of the process or when the actual process execution differs significantly from the businesses' anticipated behaviour. Process mining generally aims to describe frequent process paths (using process discovery), perform the comparison between actual processes to discovered models (using conformance analysis), and improve the incumbent business process (using process enhancement) [3]. Process prediction techniques extend the outcomes of process mining by predicting the remaining portion of the process at any given phase of its current live execution. Process mining significantly improves process prediction by providing obtained knowledge about the behaviour of the events in the process and applying this to the inference prediction model [4]. The variations of actual process executions from their standard or descriptive behaviour are assessed through conformance checking [5]. As an outcome, business users can estimate whether their business processes are running as planned or, at the very least, as stated in a documented standard process model.

In this paper, we have first used trace clustering to segregate different types of process executions in the form of sub-logs, then each sub-log is used to extract the trace-level

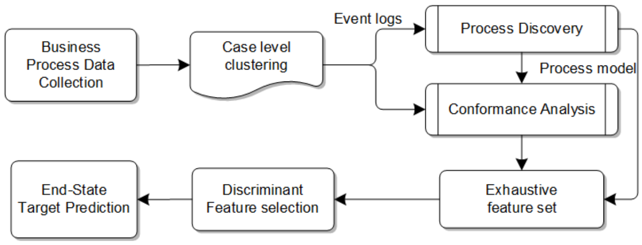

Fig. 1. High level architecture of the end-state predictor

discriminating information, referred to as features. These features are extracted using process mining for the prediction of the targeted outcome of an executing business process. *Process discovery* and *conformance analysis* techniques identify patterns in running process that characterize its end-state (outcome). These patterns also help in online prediction using the trace behaviour observed in the executing process instance. We developed a framework for a real-time process outcome prediction that uses generic trace level features extracted using the process mining techniques. *Process discovery* is used to build a graph-based model of the process and we use graph-based features from this model, and *conformance analysis* evaluates quality of alignment and control flow related features. These features provide deeper insights about execution of the business process which are difficult to identify intuitively in real-world business environment. We tested our approach using several well known machine learning algorithms, such as Naive Bayes (NB), Logistic regression (LR), Random Forest (RF) and Support Vector Machine (SVM). Fig.1 shows the high level framework of our proposed solution.

Our key contributions is the application of process mining techniques in the generation of generic features that can forecast the future outcome of a running business process. We enhanced the understanding of the significance of alignment and control flow in process prediction. We also showcase the applicability of this technique using a real-world business process case study.

The remainder of this paper is organised as follows: related work is presented in Section II, our proposed framework is presented in Section III, Section IV presents the results of the experiments along with the discussion. Section V presents the comparative analysis and Section VI concludes the presented work.

II. Literature Review

Process mining originated towards the start of the 21st century with the research and development of several algorithms for *Process Discovery*, such as the alpha algorithm, heuristic-miner algorithm and ILP (Integer Linear Programming), proposed to discover process models from given event log [2]. Alongside process discovery, performance evaluation of events and their behaviours in a given set of traces emerged as a key study area within the realm of process mining termed as *Conformance Analysis* [3]. Though process mining promises deeper analysis of executing processes, handling real-world business process is a challenging task. Tariq ar al. [6] proposed a hierarchical clustering based solution to break down large process logs into manageable smaller sub logs. Another technique, *trace clustering*, is proposed by the De Weerdt et al. [7] which divides the event log into manageable homogenous subsets.

Conformance checking techniques compare an event log with the discovered business process model, generally presented as Petri nets [8]. Several algorithms are proposed in the literature for this comparison, such as log replay algorithms by Carmona et al. [9] and trace alignment algorithm by van der Aalst [3] . We used both of these algorithms through the Process Mining tool ProM [10] to extract generic features which can present the behaviour of the instance for a given segment of a log.

Several process end-state prediction techniques are mentioned in the literature, but these techniques mostly rely on the graph-based patterns observed in the business process log, such as in [11] and [12]. Our research goal is comparable to those published by Khan et al in [13], where the authors developed a methodology for predicting a customer order process based on the individual customer's trace, but mainly focused on the graph-related features. Unlike other techniques, we have developed a novel framework for predicting the future outcomes of a running process based on both graph-based and conformance related features.

III. Proposed Framework for End-State Prediction of a Business Process

This section presents the framework for the end-state prediction of an executing business process through a real-world case study. The detailed framework of our approach is shown in Fig.2. Customers using different telecommunication services access a pre-defined business process for the resolution of their service level queries. The historic event log of this process provided valuable information about the outcome of the process, and is used as a benchmark for the identification of process end-state as either successful (customer's problem is resolved) or failed (customer's service could not be restored).

A. Data collection

We collected the data generated with the execution of the Customer Diagnostic Process (CDP) at BT, one of the UK's largest telecommunications firms. Customers of BT use a semi-automated chat facility to contact BT for solutions to their service-related queries. A predefined set of system-generated questions are used to conduct diagnostics for the customers. The flow of the questions is variable and solely dependent upon the answers given by the customer. During the process execution, the system identifies the possible solution to resolve the service complaint. If customers get their issues resolved during the call then this process execution

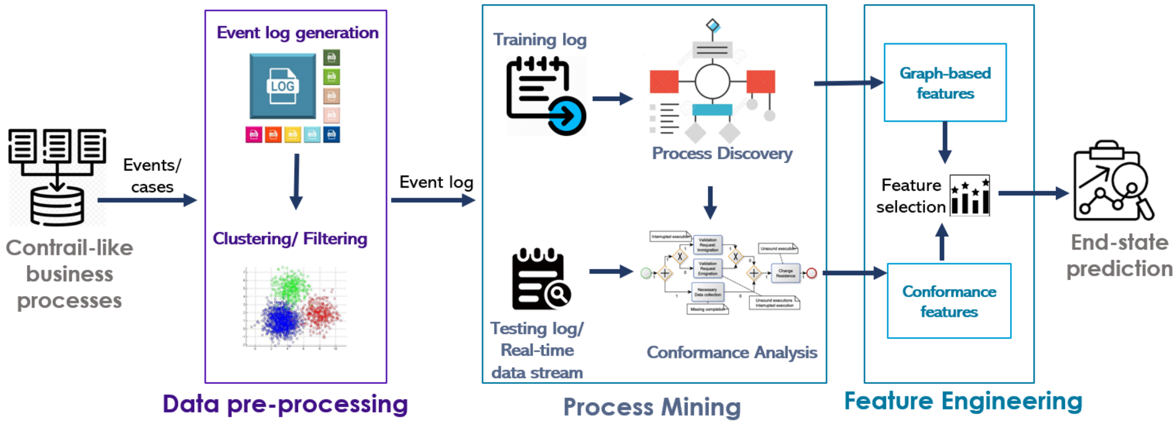

Fig. 2. Detailed framework for the process mining based end-state predictor of the business process

is regarded as successful, else identified as failed. Failure of the process may have a direct impact on the customer satisfaction, or even customer loss in extreme cases. Each customer's engagement with CDP is referred to as a process instance and represented by a unique Case ID, and questions are referred to as events. In Fig.3, the generic process is shown with the process start and end point.

For our experiments, we used 400 cases with 64 event

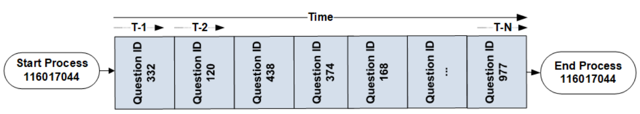

Fig. 3. Sample flow of CDP with abstracted event labels

classes, including a variety of customers types including broadband, PSTN and TV. We have an end-state labelled event log presented as 1 for successful cases and 0 for failed cases. We use 80 percent of cases for training the machine learning algorithms and 20 percent of cases are used for testing. For simplicity, we used balanced classes for both 1 and 0 labelled cases. Table I summarizes the event log used for experiments, where subcategories are identified through ActiTrac clustering, presented in Section III-D.

TABLE I
DESCRIPTION OF THE EVALUATION METRICS

Subcategory	% of cases in the process log	% of complete cases	# cases used for experiments
Broadband cases	37%	89%	204
PSTN cases	33%	90%	116
TV cases	17%	75%	48
Other cases	13%	80%	32

B. Event log presenting Contrail-like model

Organizations often update and optimize business processes regularly to cope with changing market dynamics. These updates cause deviations from the standard process; as a result, some sections of the process are generally constant and show a standard sequence, while the remaining sections contain events that are highly variable and includes fuzziness. Processes depicting these contradicting behaviours are termed as *Spaghatti processes* and *lasagna processes* by authors of [14]. Real-world business processes are variable and contain heterogeneity which makes them Contrail business processes, as proposed by Tariq et al. [6]. Most real-world business processes have elements of both Spaghetti and Lasagne processes. The evolving nature of business in a real-world business context is mainly accountable for this variability in the process. The flow of events in CDP presents Contrail-like behaviour where spaghetti-ness and lasagna behaviour is present in a single process instance, as shown in Fig.4. Constant and pre-defined system-generated groups of events are represented as M, and variable human agent generated groups of events are shown as H. Length and duration of M remains constant in each case while H shows variance in event properties.

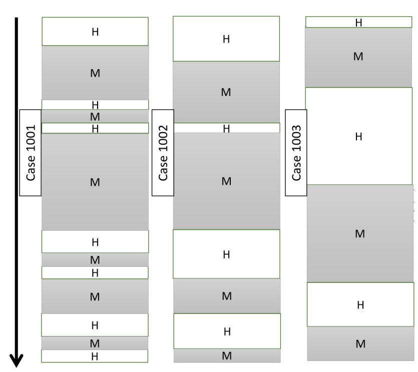

Fig. 4. Flow of events at Customer Diagnostic Process

C. Pre-processing of data

From the total of 637 cases, only 400 completed cases with target end-status Ts are considered, the rest of the cases are either incomplete or target status not labelled. Generic features are extracted from the given process log and divided into several segments bases upon their execution cut-off time. The cut-off time refers to the proportion of time that we consider as an active phase of the case to predict the future outcome. We make predictions based on feature statistics up to the active phase and compare them to the feature values till the same cut-off time of the target class. We used the cut-off times (C_{ft}) of 25%, 50%, 75%, 90% and 100% from the start time *St* till the end time *Et*, and the target end-state is attached with each trace segments *Ts*. Fig.5 shows the cut-off times mentioned in comparison to the total time $(Et - St)$. In the rest of the paper, the cut-off times 25%, 50%, 75%, 90% and 100% are expressed as a percentage of the process executed by the cut-off time percentile. We used Eq.1 to calculate the cut-off times C_{ft} for prediction of process end-state at different steps of process execution, where *Thr%* is a time threshold and T_{et} is target end time and S_t is the process start time.

$$C_{ft} = (T_{et} - S_t) * Thr\% \tag{1}$$

D. Trace clustering

The event log from a contrail-like business processes is complex and difficult to handle as it contains several interrelated process traces. Clustering of these traces allows the large process log to be segregated into several sub-logs. This facilitates the process discovery by only allowing traces with similar behaviour to be considered for analysis at a given time. We used ActiTrac, proposed in [7], to discover the clusters in the CDP log. The raw process log is clustered into broadband customers log, PSTN customers log and Tv customers log, as shown in Table I. Event data for all the process traces/cases is not completely captured in the system. It can be due to several reasons such as the customer dropped the call, or the system didn't capture all the events. We only considered completed cases with the standard process starting point and endpoint. The ProM tool is used to convert event logs into XES files and we used the AcTitraC clustering plug-in with default parameters to discover clusters among the cases.

E. Feature Extraction using process mining

In this phase, a set of generic features that characterize the behaviour of an executing process instance are extracted from the training log [2]. First, we used process discovery to identify a reliable behaviour-representing process model. The discovered model aided in the computation of graph-based

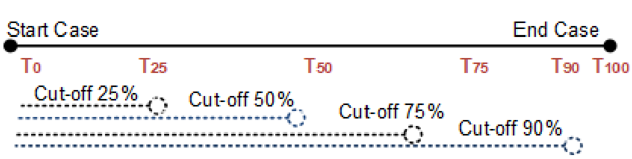

Fig. 5. Cut-off times used for end-state prediction

TABLE II
FEATURES EXTRACTED THROUGH PROCESS MINING

Feature related	Details	Symbol
Graph-based	Total number of events	Ev_total
	Number of unique events	Ev_unique
	Number of repeated events	Ev_repeated
	Number of repetitions	Re_Repatitions
	Last event time	La_event
	First event time	Fi_event
	Total duration	Ev_dur
	Duration of repeated tasks	Ti_repeated
	Number of event classes	Ev_classes
Conformance related	Fitness	f
	Precision	p
	Number of moves observed in the trace	N_m
	Number of moves possible in the log	N_mp
	% Event violations	Ev_V%
	Number of wrong events	N_EvW
	Number of missing events	N_EvM

features F_{Gr}. We used the Heuristic Miner algorithm for process model discovery. For conformance related features F_{cnf}, we performed conformance analysis which determines the actuality of synchronization between the recorded logs and the discovered model. Features extracted from conformance analysis represent the quality of the trace alignment, such as fitness *f* and precision *p*. In addition we considered the bottle-neck and performance-related features such as percentage of event violation $Ev\_V\%$ and number of wrong events $N_{Ev}W$ in the log which are not expected in the discovered process model.
For the computation of feature vectors, we started with 100% of the time and used case start time as *St* and end time as *Et*. This means that the whole process instance with all related events is considered for feature extraction at this stage (100%). Similarly, feature vectors for other cut-off thresholds are generated.

F. Feature selection

We focused only on those features for prediction which represents the actual behaviour of each process instance. For the selection of the most discriminating features, we used the Fisher discriminant ratio *(FDR)* calculated as in Eq.2:

$$FDR_f = (\mu_Y - \mu_N)^2/(\sigma_Y^2 + \sigma_N^2) \tag{2}$$

Where, μ = mean value, and σ= data variance, *Y*= successful class, *N* = failed class, *f*=num of attributes. We looked for

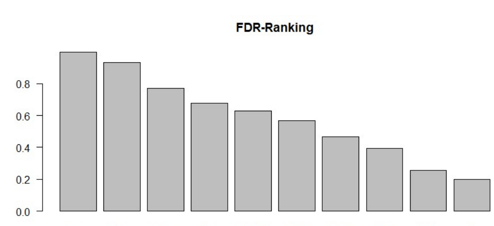

Fig. 6. Discriminant features using Fisher Discrimination Ratio

TABLE III
SAMPLE FEATURES DATA SELECTED FOR PREDICTION ANALYSIS

	Graph-based features					Conformance related features				
Case_ID	***Ev_T***	***Ev_u***	***Ev_d***	***f***	***p***	***N_m***	***N_mp***	***Ev_V%***	***N_EvW***	***N_EvM***
001	482	283	2498	0.97	24.6	1402	57470	2.3	32	150
002	160	115	153	0.89	24.3	1287	57070	2.1	38	157
003	405	253	486	0.99	22.9	1566	58070	2.1	35	148
004	504	288	345	0.73	28.1	1899	57083	3.6	65	225

the higher number of ratios as it presents the discriminating power of the feature for the prediction of the target variable. FDR analysis resulted in a total of 10 features as presented in Fig.6. The feature vector we used for classification along with sample data is presented in Table III.

IV. RESULTS AND DISCUSSION

We used the obtained multi-dimensional feature vector to predict the future outcome of the executing process instance through classification implemented using several machine learning algorithms. For all the experiments, we used Intel Core-i5-8th Gen processor at 1.80 GHz, 16GB RAM, Windows 10 Enterprise (64-bit). To analyse the predictive model, we used 10-fold cross-validation and split the original event log into a training set and a testing set. Training is done using 80% of the event log with 160 cases. For the testing, we used an event log with 5 different cut-off times (25%, 50%, 75%, 90% and 100%). For each of the cut-off times, we computed the 10 features (3 graphic-based features and 7 conformance features) as given in Table III. We used Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machines (SVM) and Random Forest (RF) algorithms to predict the end-state of the executing process instance.

As observed in Table IV, for 25% of cut-off time, Random Forest(RF) resulted in 56.17% accuracy as compared to the average accuracy of other algorithms which is 50.36%. The Area Under the Curve (AUC) is also highest for RF which is 68.17% where the lowest AUC is observed through SVM (55.04%). For cut-off times of 50% and 75%, the average accuracy of 69.21% and 76.45% is observed along with the highest AUC from SVM(83.19%) for 50% and Naïve Bayes (82.19%) for 75% time. Due to the contrail-like behaviour of events observed in the log, the range of Ev_Total in 50% of cut-off is between 8 to 332 and for 75% time is between 8 to 398. Training time for thousand rows in both 50% and 75% of logs is almost identical. The total time taken to predict in both 50% and 75% of logs is highest for SVM (1110831ms) and the quickest predictor algorithm has been Naïve Bayes with a total duration of 6104ms. At the cut-off time of 90% & 100%, this is when the process has almost reached its end and predictor results in the accuracy of 93% and 100% respectively, showing that both classes (0&1) are highly predicted at the closing stage of the process. Although, the computation time required by algorithms is highest at this stage, for instance, the scoring time of SVM is 9171ms at 90% of cut-off time, which is 30 times higher than SVM time for 25% cut-off time.

V. COMPARISON WITH OTHER PROCESS END-TARGET PREDICTION TECHNIQUES

For the evaluation of our proposed techniques, we compared our approach's accuracy and F-measure with the technique proposed in [13] by taking average scores of accuracy and F-measure using four machine learning algorithms, i.e. NB, LR, RF & SVM. Khan et al. [13] used only graph-based features extracted from an event log of the real-world dataset and predicted the outcome of the process as early as 25% of the process time. As shown in Fig.7, our technique of using augmented features, i.e. conformance and graph-based, (*CGrF*) outperformed in both the accuracy and F-measure when only graph-based features were used. Our technique

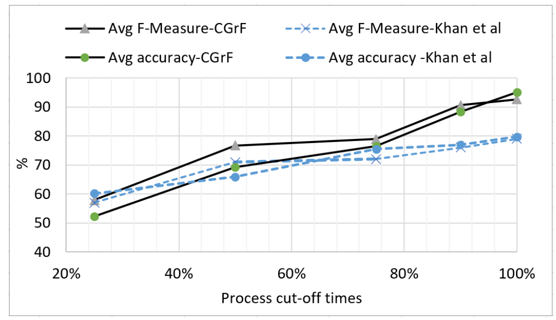

Fig. 7. Comparison of average accuracy and average F-measure

showed the lowest average accuracy of 52.1% at 25% of the process execution which is around 7% less compared to [13], this is due to the contrail behaviour of the event log we used for experiments. 80% of the events at the start of the process (up to 25% cut-off) are machine-generated events and do contribute much to outcome prediction at this level. As a process executes, the control-flow perspective dominates and conformance analysis results in generating more discriminating features. Our technique showed an accuracy of 88% at the 90% cut-off, which is 12% higher than the prediction by [13]. Similarly, F-measure comparison is presented in Fig.7 with 57.87%, 76%, 79%, 91% and 93% as compared to the 57%, 71%, 72%, 76% and 79% score of [13].

VI. CONCLUSION

Processes in the real-world business environment are complex and need to be monitored for the earliest possible prediction of the process outcome as it is an essential factor for improved customer experience. Also, early identification of the process end-state allows business users to proactively make decisions to mitigate possible business loss. To overcome this challenge, our paper presents a novel solution which computes highly discriminating features from live real-world business process. These features are used to predict the future outcome of the executing process through machine learning. We augmented process mining techniques to extract features after the initial clustering of cases and then tested the prediction at four stages of the process, referred to as cut-off times of process and showed that our solution is very

TABLE IV
PREDICTION RESULTS FOR THE PROCESS END-STATE USING SEVERAL MACHINE LEARNING ALGORITHMS

			25% Cut-off time				
Prediction Algorithm	**Accuracy %**	**AUC %**	**F-Measure %**	**Recall %**	**Total Time (ms)**	**Training Time (1,000 Rows)**	**Scoring Time (1,000 Rows)**
Naive Bayes	53.52%	63.52%	61.52%	81.52%	4908	188	225
Logistic Regression	53.52%	63.52%	62.52%	82.52%	1223	203	250
Random Forest	56.17%	68.17%	63.17%	83.17%	4988	60	319
Support Vector Machine	50.04%	55.04%	51.04%	71.04%	6104	183	331
			50% Cut-off time				
Prediction Algorithm	**Accuracy %**	**AUC %**	**F-Measure %**	**Recall %**	**Total Time (ms)**	**Training Time (1,000 Rows)**	**Scoring Time (1,000 Rows)**
Naive Bayes	69.19%	82.19%	78.19%	88.19%	6104	7	52
Logistic Regression	69.30%	75.30%	73.30%	80.30%	6053	14	56
Random Forest	69.08%	79.08%	74.08%	81.08%	21165	12	164
Support Vector Machine	69.19%	83.19%	79.19%	84.19%	1110831	3484	6407
			75% Cut-off time				
Prediction Algorithm	**Accuracy %**	**AUC %**	**F-Measure %**	**Recall %**	**Total Time (ms)**	**Training Time (1,000 Rows)**	**Scoring Time (1,000 Rows)**
Naive Bayes	79.00%	82.00%	81.00%	82.00%	6104	7	52
Logistic Regression	79.50%	83.50%	81.50%	83.50%	6053	14	56
Random Forest	76.20%	81.20%	78.20%	80.20%	21165	12	164
Support Vector Machine	70.70%	74.70%	73.70%	77.70%	1110831	3484	6407
			90% Cut-off time				
Prediction Algorithm	**Accuracy %**	**AUC %**	**F-Measure %**	**Recall %**	**Total Time (ms)**	**Training Time (1,000 Rows)**	**Scoring Time (1,000 Rows)**
Naive Bayes	93.00%	96.80%	95.60%	95.86%	16356	16	124
Logistic Regression	89.50%	92.90%	91.60%	92.44%	15522	48	145
Random Forest	90.20%	93.10%	91.70%	92.46%	40873	28	398
Support Vector Machine	81.70%	85.90%	84.30%	84.48%	1219542	3374	9171
			100% Cut-off time				
Prediction Algorithm	**Accuracy %**	**AUC %**	**F-Measure %**	**Recall %**	**Total Time (ms)**	**Training Time (1,000 Rows)**	**Scoring Time (1,000 Rows)**
Naive Bayes	95.00%	98.70%	97.30%	97.46%	22686	23	140
Logistic Regression	90.50%	93.90%	92.40%	93.08%	18898	54	148
Random Forest	92.20%	96.00%	94.90%	95.11%	48644	45	416
Support Vector Machine	83.70%	87.20%	86.10%	86.43%	1222734	3387	9179

promising for the earliest prediction. In future, we will extend this work to include end state prediction based on event-level business driven process segments.

ACKNOWLEDGMENT

This research is supported by the BTIIC (BT Ireland Innovation Centre) project, funded by BT and Invest Northern Ireland.

REFERENCES

[1] W. Kratsch, J. Manderscheid, M. Röglinger, and J. Seyfried, "Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction," *Business & Information Systems Engineering*, pp. 1–16, 2020.

[2] P. N. Taylor, "Customer contact journey prediction," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 2017, pp. 278–290.

[3] W. Van Der Aalst, "Data science in action," in *Process mining*. Springer, 2016, pp. 3–23.

[4] D. Ruta and B. Majeed, "Business process forecasting in telecom industry," in *2011 IEEE GCC Conference and Exhibition (GCC)*. IEEE, 2011, pp. 389–392.

[5] A. Rozinat and W. M. Van der Aalst, "Conformance checking of processes based on monitoring real behavior," *Information Systems*, vol. 33, no. 1, pp. 64–95, 2008.

[6] Z. Tariq, N. Khan, D. Charles, S. McClean, I. McChesney, and P. Taylor, "Understanding contrail business processes through hierarchical clustering: A multi-stage framework," *Algorithms*, vol. 13, no. 10, p. 244, 2020.

[7] J. De Weerdt, S. Vanden Broucke, J. Vanthienen, and B. Baesens, "Active trace clustering for improved process discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 12, pp. 2708–2720, 2013.

[8] F. Mannhardt, M. De Leoni, H. A. Reijers, and W. M. Van Der Aalst, "Balanced multi-perspective checking of process conformance," *Computing*, vol. 98, no. 4, pp. 407–437, 2016.

[9] F. Taymouri and J. Carmona, "A recursive paradigm for aligning observed behavior of large structured process models," in *International Conference on Business Process Management*. Springer, 2016, pp. 197–214.

[10] H. Verbeek, J. Buijs, B. Van Dongen, and W. M. van der Aalst, "Prom 6: The process mining toolkit," *Proc. of BPM Demonstration Track*, vol. 615, pp. 34–39, 2010.

[11] J. Maeyens, A. Vorstermans, and M. Verbeke, "Process mining on machine event logs for profiling abnormal behaviour and root cause analysis," *Annals of Telecommunications*, vol. 75, no. 9, pp. 563–572, 2020.

[12] C. Rinner, E. Helm, R. Dunkl, H. Kittler, and S. Rinderle-Ma, "Process mining and conformance checking of long running processes in the context of melanoma surveillance," *International journal of environmental research and public health*, vol. 15, no. 12, p. 2809, 2018.

[13] N. Khan, Z. Ali, A. Ali, S. McClean, D. Charles, P. Taylor, and D. Nauck, "A generic model for end state prediction of business processes towards target compliance," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer, 2019, pp. 325–335.

[14] W. Van Der Aalst, A. Adriansyah, A. K. A. De Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van Den Brand, R. Brandtjen, J. Buijs *et al.*, "Process mining manifesto," in *International conference on business process management*. Springer, 2011, pp. 169–194.